

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A computer-implemented method of sharing markup text from a page among a plurality of users ~~compiling a page containing markup text into an application that outputs markup in response to a requests from a said plurality of users~~, said method comprising:
analyzing the page to extract markup text, wherein the page includes at least the markup text and a set of code instructions to be executed as an application;
pre-initializing a static variable of a class to contain the markup text from the page; and
loading the class containing the pre-initialized static variable into a shared, read-only memory, wherein the shared, read-only memory is accessible to said plurality of users.
2. (Currently amended) A The computer-implemented method according to claim 1, further comprising storing the markup text in a resource file associated with the application.
3. (Canceled)
4. (Canceled)

5. (Currently amended) A method of initiating a first instance of an application that shares a set of markup text with other instances of the application, wherein the first instance of the application is generated by compiling a code from a page that contains both the code and the set of containing markup text into an application that outputs markup in response to a request from one or more a users, said method comprising:

generating executing instructions for to instantiate the first instance of the application, ~~before~~ wherein said instructions are stored on a computer-readable medium, said instructions that, when executed, cause one or more processors to perform the steps of:

analyzing said page to distinguish between the code and the set of
markup text;

pre-initializing a class containing a static variable to contain the set
of markup text;

~~in one time the application is executed, loading a~~ the class containing a the static variable into a shared, read-only memory, wherein said shared, read-only memory is accessible to the one or more users, said static variable being pre-initialized to contain the markup text; and

~~in a subsequent time the application is executed, accessing the markup~~ text in the shared, read-only memory when the code from the first instance of the application is executed.

6. (Currently amended) A method according to claim 5, wherein the class is not loaded into the shared, read-only memory when the other instances of the ~~in the subsequent time the application is~~ are executed.
7. (Canceled)
8. (Canceled)
9. (Previously Presented) A method according to claim 1, wherein:
the markup text includes information to be displayed to a user and an annotation directing a user agent how to render the information to be displayed to the user; and
the markup output by the application includes the annotation.
10. (Previously Presented) A method according to claim 1, wherein the static variable of a class is an array of characters.
11. (Currently amended) A method according to claim 5, wherein:
the set of markup text includes information to be displayed to a user and an annotation directing a user agent how to render the information to be displayed to the user; and
the set of markup output by the application includes the annotation.

12. (Previously Presented) A method according to claim 5, wherein the static variable of a class is an array of characters.
13. (Canceled)
14. (Canceled)
15. (New) A computer-readable storage medium bearing instructions that, when executed, cause one or more processors to perform a method for sharing markup text from a page among a plurality of users in response to requests from said plurality of users, said method comprising:
analyzing the page to extract the markup text, wherein the page includes at least the markup text and a set of code instructions to be executed as an application;
pre-initializing a static variable of a class to contain the markup text from the page; and
loading the class containing the pre-initialized static variable into a shared, read-only memory, wherein the shared, read-only memory is accessible to said plurality of users.
16. (New) The computer-readable storage medium of claim 15, further comprising instructions to store the markup text in a resource file associated with the application.

17. (New) The computer-readable storage medium of claim 15, wherein:
the markup text includes information to be displayed to a user and an annotation
directing a user agent how to render the information to be displayed to the
user; and
the markup output by the application includes the annotation.
18. (New) The computer-readable storage medium of claim 15, wherein the static
variable of a class is an array of characters.
19. (New) A computer-readable storage medium bearing instructions that, when
executed, cause one or more processors to perform a method of initiating a first
instance of an application that shares a set of markup text with other instances of
the application, wherein the first instance of the application is generated by
compiling code from a page that contains both the code and the set of markup text
in response to a request from one or more users, said method comprising:
executing instructions to instantiate the first instance of the application;
analyzing said page to distinguish between the code and the set of markup text;
pre-initializing a class containing a static variable to contain the set of markup
text;
loading the class containing the static variable into a shared, read-only memory,
wherein said shared, read-only memory is accessible to the one or more
users; and
accessing the markup text in the shared, read-only memory when the code from
the first instance of the application is executed.

20. (New) The computer-readable storage medium of claim 19, wherein the class is not loaded into the shared, read-only memory when the other instances of the application are executed.
21. (New) The computer-readable storage medium of claim 19, wherein:
the set of markup text includes information to be displayed to a user and an annotation directing a user agent how to render the information to be displayed to the user; and
the set of markup output by the application includes the annotation.
22. (New) The computer-readable storage medium of claim 19, wherein the static variable of a class is an array of characters.